

In my old age I've mostly given up trying to convince anyone of anything. Most people do not care to find the truth, they care about what pumps their bags. Some people go as far as to believe that *perception is reality* and that truth is a construction. I hope there's a special place in hell for those people.

It's why the world wasted \$10B+ on self driving car companies that obviously made no sense. There's a much bigger market for truths that pump bags vs truths that don't.

So here's your new truth that there's no market for. Do you believe a compiler can code? If so, then go right on believing that AI can code. But if you don't, then AI is no better than a compiler, and arguably in its current form, worse.

The best model of a programming AI is a compiler.

You give it a prompt, which is "the code", and it outputs a compiled version of that code. Sometimes you'll use it interactively, giving updates to the prompt after it has returned code, but you find that, like most IDEs, this doesn't work all that well and you are often better off adjusting the original prompt and "recompiling".

While noobs and managers are excited that the input language to this compiler is English, English is a poor language choice for many reasons.

1. It's not precise in specifying things. The only reason it works for many common programming workflows is because they are common. The minute you try to do new things, you need to be as verbose as the underlying language.
2. AI workflows are, in practice, highly non-deterministic. While different versions of a compiler might give different outputs, they all promise to obey the spec of the language, and if they don't, there's a bug in the compiler. English has no similar spec.
3. Prompts are highly non local, changes made in one part of the prompt can affect the entire output.

tl;dr, you think AI coding is good because compilers, languages, and libraries are bad.

This isn't to say "AI" technology won't lead to some extremely good tools. But I argue this comes from increased amounts of search and optimization and patterns to crib from, not from any magic "the AI is doing the coding". You are still doing the coding, you are just using a different programming language.

That anyone uses LLMs to code is a testament to just how bad tooling and languages are. And that LLMs can replace developers at companies is a testament to how bad that company's codebase and hiring bar is.

AI will eventually replace programming jobs in the same way compilers replaced programming jobs. In the same way spreadsheets replaced accounting jobs.

But the sooner we start thinking about it as a tool in a workflow and a compiler—through a lens where tons of careful thought has been put in—the better.

I can't believe anyone bought those vibe coding crap things for billions. Many people in self driving accused me of just being upset that I didn't get the billions, and I'm sure it's the same thoughts this time. Is your way of thinking so fucking broken that you can't believe anyone cares more about the *actual truth* than make believe dollars?

[From this study](#), AI makes you *feel* 20% more productive but in reality makes you 19% slower. How many more billions are we going to waste on this?

Or we could, you know, do the hard work and build better programming languages, compilers, and libraries. But that can't be hyped up for billions.

Update: I added some follow up as a [reply to this comment](#) on HN. I'm not against AI at all, I'm pro recognizing its strengths and limitations as a tool, and once you do that, if it makes sense for your workflow by all means use it. I also think AI will continue to improve at a careful and steady pace.

What I'm against is hype.

Which brings me to a meta point about this blog. If I wrote in a more careful and technical style, it would get a lot less traction because it would be